Pazmiño-Maji, R. A., García-Peñalvo, F. J., & Conde-González, M. Á. (2017). Comparing Hierarchical Trees in Statistical Implicative Analysis & Hierarchical Cluster in Learning Analytics. In J. M. Dodero, M. S. Ibarra Sáiz, & I. Ruiz Rube (Eds.), *Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17) (Cádiz, Spain, October 18-20, 2017)* (Article 49). New York, NY, USA: ACM. doi:10.1145/3144826.3145399

Comparing Hierarchical Trees in Statistical Implicative Analysis & Hierarchical Cluster in Learning Analytics

Rubén A. Pazmiño-Maji
CIDED Research Group,
Science Faculty, Escuela Superior Politécnica de Chimborazo
060155 Riobamba, Ecuador
rpazmino@espoch.edu.ec

Francisco J. García-Peñalvo
GRIAL Research Group,
Research Institute for Educational Sciences, University of Salamanca
37008 Salamanca, Spain
fgarcia@usal.es

Miguel A. Conde-González
Department of Mechanical, Computer Science and Aerospace Engineering,
University of León
24071 León, Spain
miguel.conde@unileon.es

ABSTRACT

Learning Analytics has been and is still an emerging technology in education; the amount of research on learning analysis is increasing every year. The integration of new open source tools, analysis methods, and other calculation options are important. This paper aims to compare hierarchical trees in Statistical Implicative Analysis (SIA) and some hierarchical clusters in Learning Analytics. To this end, we must use a quasi-experimental design with random binary data. A comparison is about the time it takes to evaluate the function for execute the four cluster algorithms: cohesion tree (ASI), similarity tree (ASI), agnes (cluster R package) and hclust (R base function). This paper provides an alternative hierarchical cluster used in Statistical Implicative Analysis that is possible to use in Learning Analytics (LA). Also, provides a comparative R-program used and identifies future research about software performance.

CCS CONCEPTS

• Information systems → Clustering • Software and its engineering → Software performance • Information systems → Open source software

KEYWORDS

Clustering, Software performance, Learning analytics, statistical implicative analysis, Open source software, hierarchical cluster, similarity tree.

1 INTRODUCTION

Horizon Report 2016 [1] specifies that learning analytics [2-4] and adaptive learning [5-7] have been and are still an emerging technology. Hierarchical Cluster Analysis (HCA) is frequently used in the discovery and predictive analysis in Learning Analytics (LA). Enterprise Performance Management document [8] about LA, classified the organizations in three generation (G): G1 is about descriptive and partially diagnostic, here are the 90% of organizations. G2 is about partially diagnostic, discovery and partially predictive, here are between 5 and 9% of organizations. However, G3 is about partially Predictive and prescriptive; here there are not organizations.

The analytics maturity model proposed by Bichsel is used to evaluate the progress in academic and learning analytics. In the advances they have produced positive results but, most institutions also scored low for data analytics tools, reporting, and expertise [9]. In this paper, we propose a quick data analytic tool frequently used in SIA but not in LA. Also, a priority task with the analysis methods used in Data Mining and Learning Analytics is to analyze precision, accuracy, sensitivity, coherence, fitness measures, cosine, confidence, lift, similarity weights [10] and of course speed, for optimizing and adapt them.

The principal aim of this paper is to determine and compare the processing time of four cluster algorithms: cohesion tree, similarity tree, agnes and hclust.

In section 2, we present a proximation to LA, agnes, hclust, SIA, cohesion tree, and similarity tree. In section 3, show in detail the quasi-experimental design, process, software and hardware used. In seccion 4 show the results and its discussion. In Section 5 we show the conclusions and future works.

2 HIERARCHICAL CLUSTERING

The Baker and Inventado book [11], classify de analysis methods used in learning analytics based in educational data mining: Classification, Regression, Latent Knowledge Estimation, Association Rule Mining, Sequential Pattern Mining, Correlation Mining, Causal Data Mining, Clustering, Factor Analysis, Domain Structure Discovery and Discovery with models. The paper Statistical implicative Analysis approximation to Learning Analytics [12], show us the principals SIA analysis methods used
In LA are: Clustering (37.5%) and Association Rule Mining (95.8%). The paper [10] also shows us the great use of cluster methods in Learning Analytics. Papers comparing SIA and LA or data mining methods are scarce. In the following paragraphs, we shown some papers that work with SIA and other cluster methods.
The next research is about a comparison between hierarchical clustering of variables, implicative statistical analysis and confirmatory factor analysis. A study was made in a school with students in 6 grades to get some data about apprehension of geometrical figures through a comparison between hierarchical clustering of variables, implicative statistical analysis and confirmatory factor analysis. The purpose of this study is to clarify the features and benefits of applying these three methods by comparing the results of their application in the functional learning of the geometric figures. The findings of the study suggest that the three statistical methods are open to complementary use and each one does not operate at the expense of the other, confirmatory factor analysis provided a means to give meaning to the structure of operative apprehension of geometric figures. Hierarchical grouping of variables is a means to classify student responses, to identify student's consistencies and inconsistencies between different conversions, and to investigate factors influencing this behavior. The implied method provided a means to examine the relationship between task responses and the relative difficulty of different conversions based on student performance [13]. A first attempt to use the SIA analysis tools in the cluster analysis is clustering medical images [14], the CHIC software and the reduction option was used. Interesting results were obtained, validated by experts in the treatment of medical images. A comparative analysis on the visual perception of the hierarchical cluster of 63 simple images was made. 35 students took part in the experiment. Approximately 70% of the students agree or strongly agree with the groups created with the implicate statistical analysis [15].
In the following subsection, we show the main hierarchical cluster methods used in LA and the two graphics trees used in SIA.

2.1 LA and Hierarchical Cluster

Clustering are methods used to classify subjects (or variables) within a data set, into a multiple group based on their similarity. Hierarchical cluster does not require to pre-specify the number of clusters to be produced. Hierarchical cluster has a graphic result called dendrogram, which is like an inverted tree. There are two groups of Hierarchical Cluster Agglomerative or agglomerative nesting (Agnes – Fig. 1) and divisive or divisive analysis (Diana) [16].

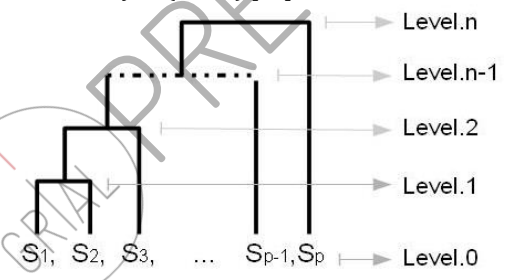

Figure 1: Agglomerative clustering and hierarchical levels

In agglomerative clustering, each observation is initially considered as a cluster of its one (level.0). Then, the most similar clusters are successively merged (level.1, level.2, …., level.n) until there is just one single big cluster (level.n). Agglomerative clustering uses a bottom-up manner.
"R is a free software environment for statistical computing and graphics" [17]. Cluster is a R package (2017-03-10) that allows to perform cluster analysis in R. It provides the function agnes() for computing agglomerative computing [18]. The R base function hclus() can be used to create the hierarchical tree. There are many cluster agglomeration methods. The most common are maximun, minimun, mean, centroid linkage, ward's. The last cluster agglomeration method is generally preferred [18].

2.2 SIA and Hierarchical Trees

Fig. 2 shows the automatic classification of the fungus Suillus granulatus, based on the intensity of the enzymatic browning and generated in the form of an similarity tree based on the concept of Lerman similarity [19].
The tree of similarity calculates for each pair of variables the similarity between them. Then, add the classes themselves made up of other classes.

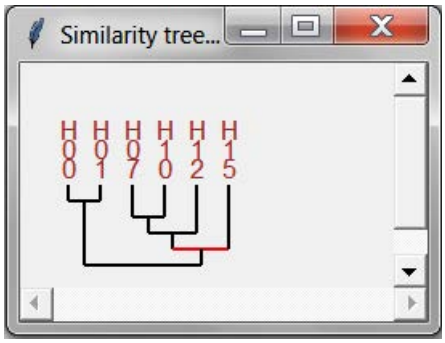

Figure 2: **Similarity tree in Rchic Software**

In the tree presented in Fig. 2, the variables H00 and H01 are the most similar, as well as H07 and H10, in the case of H07 and H10 the algorithm allows to create new classes like ((H07, H10), H12). Finally, in the last level, we obtain a single class (((H07, H10), H12), H15), the levels identified by a red line are significant levels insofar as they have higher classification significance than the other levels.

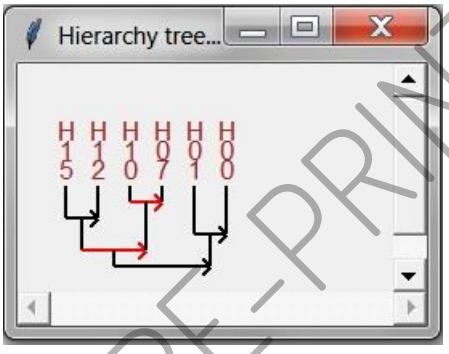

Figure 3: **Hierarchy tree in Rchic Software**

Fig. 3 shows the automatic classification of the fungus Suillus granulatus, based on the intensity of the enzymatic browning and generated in the form of an asymmetric hierarchical tree based on the concept of cohesion [20]. The graphic generated in Fig. 3, shows at the top the labels that identify the 6 images. The vertical lines classify the images, so for example 3 groups are formed by 2 images: H10 and H07 that are in the same group with the greater similarity (to be higher), as well as H15 and H12 that have a second group of similarity and, finally, H01 and H00 that form a third group of similarity (being lower down between the groups of two images). It has a single group conformed by the four images: H15, H12, H10 and H07. Finally, the largest group made up of the 6 images. In total there were 5 groups, 3 of two images, 1 group consisting of 4 images and 1 group with all 6 images. Horizontal lines are arrows that give additional information about the direction of the grouping. Arrows in red are significant nodes. Using the Rchic [21, 22] options, you can continue generating and exploring different hierarchical trees, just activate the variables that you want to study and disable those that you do not want to include in the study. This option allows us to generate dynamic environments of the type "What happens?", Which can be explored to arrive at classifications of our greatest interest.

3 METHODOLOGY

In this section, we show in detail the quasi-experimental design, graphical process, R-program, R-functions, R-packages, other software and hardware used.

3.1 Process

The process followed in the quasi experiment is shown in Fig. 4. the process had four main stages: Quasi-experimental design, Computational details and R Programming, Quasi experiment execution and Data Analysis. The stages were consecutive, dependent and complementary.

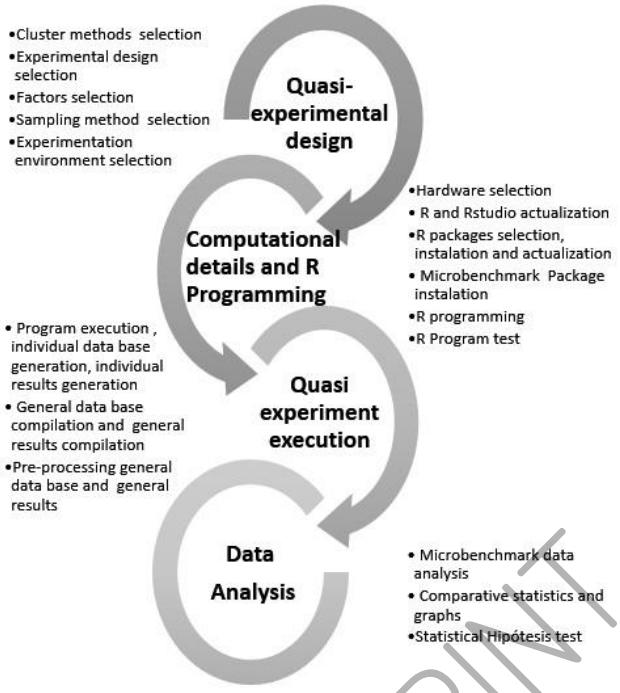

Figure 4: **Process followed in the quasi-experiment**

3.2 Quasi-experimental design

To demonstrate the following hypothesis:

H_0: The algorithms are identical (in terms of time it takes to evaluate the function for execute algorithms)

H_1: At least a couple of algorithms are different (in terms of time it takes to evaluate the function for execute algorithms)

a quasi-experiment of the RGXO [23, 24] type was proposed, where the representative of the selection was random, G represent the groups with 3 replicates (automatically generated by the microbenchmark program), X is the treatment (cohesion tree, tree of similarity, agnes and hclust) and O represents the post-test that was the measurement of time.

After checking the assumptions of independence, normality and homoscedasticity, a non-parametric Kruskal–Wallis test [25] was used with a single factor: the 4 hierarchical cluster methods (cohesion tree, similarity tree, agnes and hclust), a level of significance of 95%. The dependent variable is the execution time (in seconds) that is of numeric type. The population size is 100000 databases formed up to a maximum of 1000 observations and 100 variables. For its size, a sample was chosen using the simple random sampling method with parameter of interest the mean, the following formula [26] was considered for the calculation of the sample size.

$$n = \frac{s^2}{\frac{E^2}{Z^2_{\frac{\alpha}{2}}} + \frac{s^2}{N}}$$

For the application of the formula the following parameters were used:
Standard deviation = 1;
α = 5%;
Z = 1.96;
E = 10%;
N = 100000
the sample size was 382.675, which is approximately 383 random binary data bases.

3.3 Computational details and R programming

A computer with an Intel® Core ™ i7-4770 CPU @ 3.40ghZ 3.40 GHz microprocessor was used for the study. We worked with the free

statistical software R version 3.4.1 and the free integrated development environment (IDE) RStudio version 1.0.153 [27]. The cohesion tree, similarity tree functions belong to the Rchic package version o.24. The agnes function was extracted from the cluster package version 2.06. The hclust function is a base function of R version 3.4.1. All databases were binary and were generated randomly using the program of Figure 4 with the functions of Figure 5. Microbenchmark v1.4 [28] tries hard to accurately measure only the time it takes to evaluate a given expression. To achieved this, the nanosecond accurate timing functions most modern operating systems provide are used. The code in Fig. 5 (Functions) and Fig. 6 (R-program) were used in the execution of the quasi-experiment.

3.4 Quasi-experiment execution and data analysis

We use the R software version 3.4.1 for data analysis of the 4596 subjets and two initial variables: Methods and Time.ns. Using the code from Fig. 5 and Fig. 6, 383 files were generated in flat format *.csv. Then they joined together in a single database in Excel 2016 and the next variables were generated: VarNum, VarGrup, SubjNum, SubjGroup, DatTot, DatGroup, Replication, Time.us, and Time.s.

The data used in the quasi-experiment you can load for the next link:
https://1drv.ms/f/s!AgztBzyVpXfKgv8U3LGMMjSdSdLEnA

```
hrarchy<-function(x){
  hr<-callHierarchyTree (x,contribution.supp=FALSE,
                         typicality.supp=FALSE,
                         computing.mode=3,verbose=FALSE)
  return (hr)
}
hclus_c<-function(x){
  df<-read.csv(x, sep=";")
  res.dist <- dist(df, method = "euclidean")
  hc1      <- hclust(res.dist, method = "ward.D")
  dend1    <- as.dendrogram(hc1)
  fviz_dend(dend1, cex = 0.6,main = "Hclust")
  return (dend1)
}
agnes_c<-function(x){
  df<-read.csv(x, sep=";")
  res.agnes <- agnes(x=df,stand = TRUE,
                     metric = "euclidean",
                     method = "ward")
  fviz_dend(res.agnes, cex = 0.6,main = "Agnes")
  return (fviz_dend)
}
simlrty<-function(x){
  st<-callSimilarityTree(x,contribution.supp=FALSE,
                         typicality.supp=FALSE,verbose=FALSE)
  return (st)
}
```

Figure 5: R function for execute the four cluster algorithms

```
library("microbenchmark");
library("rchic");
library("ggplot2");
library("factoextra");
library("cluster")
hrarchy<-function(x){...}
hclus_c<-function(x){...}
agnes_c<-function(x){...}
simlrty<-function(x){...}
nprocess<-4;rep<-3;endfor<-383
TOTALtimeCompareSimilarityMatrix<-
  matrix(1:(rep*nprocess),nrow=(rep*nprocess),ncol=1)
for (i in 1:endfor){
     nVar<- round(runif(1),2)*100
     nFilas<- round(runif(1),3)*1000
     DataBase<-replicate(nVar, round(runif(nFilas),0))
     rownames(DataBase)<-paste('S',1:nFilas,sep='')
     colnames(DataBase)<-c(';V1',paste('V',2:nVar,sep=''))
     f<-paste('_D',toString(i),'.csv',sep="_")
     T<-paste('_T',toString(i),'.csv',sep="_")
     write.table(DataBase, file=f, sep=";" ,quote = FALSE)
     DataBaseT<-t(DataBase)
     write.table(DataBaseT, file = T ,sep=";")
     timeCompareSimilarity<-microbenchmark(
        hclus_c(T), agnes_c(T), simlrty(f), hrarchy(f),
        times=rep,unit="ms",control=list("inorder"))
        timeCompareSimilarityMatrix<-
          as.matrix(timeCompareSimilarity)
     TOTALtimeCompareSimilarityMatrix<-
       cbind(TOTALtimeCompareSimilarityMatrix,
             timeCompareSimilarityMatrix)
     colnames(TOTALtimeCompareSimilarityMatrix)[2*i+1]<-
       paste(toString(nVar),'x',toString(nFilas),
             paste('_D',toString(i),sep="_"))
     write.table(TOTALtimeCompareSimilarityMatrix,
                 file = "out.csv",sep=";",row.names = FALSE)
}
```

Figure 6: **R program for data bases generation**

4 RESULTS AND DISCUSSION

In this section, we show and discuss the numerical and graphical results obtained. Fig. 7 shows that there is a significant difference between the time, the time no is the same in the four algorithms.

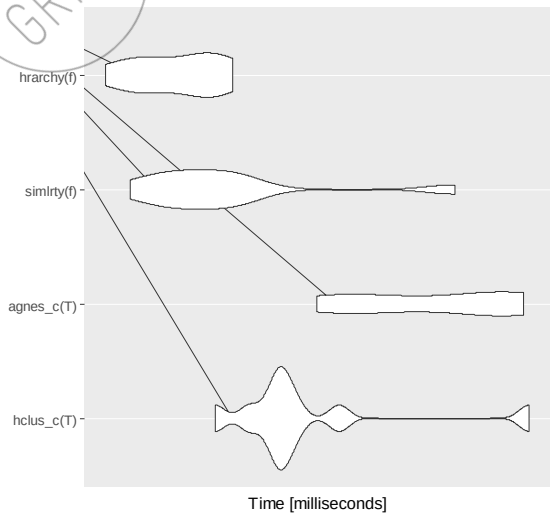

Figure 7: **Microbenchmark autoplot output**

Fig. 7 shows the same than Fig. 8: there is a significant difference between the time, but the algorithms cohesion tree and similarity tree seem similar and smaller in time than the hclust and agnes.

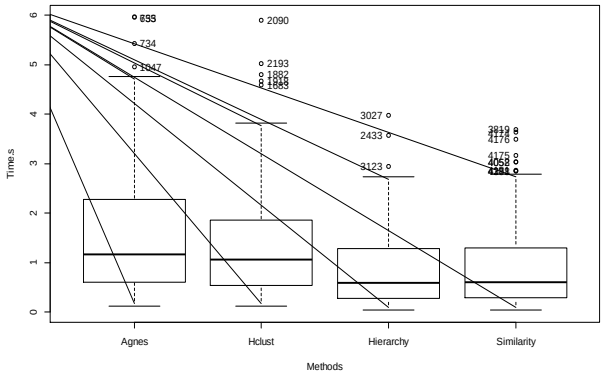

Figure 8: Comparative Box-plot

The statistical hypothesis for comparison about the time it takes to evaluate the function for executing the four cluster algorithms are:

H_0: The algorithms are identical (in terms of time it takes to evaluate the function for execute algorithms)
H_1: At least a couple of algorithms are different (in terms of time it takes to evaluate the function for execute algorithms)

First, let us analyze if we can use a PARAMETRIC STATISTICAL TEST. Let us show the assumptions of the analysis of variance (ANOVA).
1) The independence is true, because the data generation is random.
2) Homoscedasticity, the Bartlett test results are:
The Bartlett test of homogeneity of variances:
Agnes Hclust Hierarchy Similarity
1.1975861 0.7606134 0.4103789 0.4485501
data: Time.s by Methods
Bartlett's K-squared = 440.01, df = 3, p-value < 2.2e-16
Conclusion: The null hypothesis is rejected and therefore, we conclude that the homogeneity of variances is false.
3) Normality, we show it using Kolmogorov-Smirnov normality test, the result is:
Lilliefors (Kolmogorov-Smirnov) normality test
data: Time.s
D = 0.11896, p-value < 2.2e-16
Conclusion: The null hypothesis is rejected and therefore, we conclude that the sample has not been extracted from a normal population.

Because all the assumptions are not met, we must use the NON-PARAMETRIC STATISTICAL TEST of Kruskal–Wallis.
Kruskal-Wallis rank sum test
data: Time.s by Methods
Kruskal-Wallis chi-squared = 474.54, df = 3, p-value < 2.2e-16
Conclusion: The null hypothesis (using ranks) is rejected and therefore: There is a significant difference between the algorithms time. Time is not the same in at least one of the algorithms (Agnes, Hclust, Hierarchy, or Similarity). A two to two comparison is needed at the future using the R-packages dunn.test, conver.test, PCMCR, or other.

5 CONCLUSIONS

Considering all the elements indicated in this paper and with the specified functions, software, and hardware features, we can say with a level of significance of 5%. The assumptions of independence are true, because the data generation is random. We conclude that the homogeneity of variances is false (p-value < 2.2e-16). We conclude that the sample has not been extracted from a normal population (p-value < 2.2e-16). The difference between the times to evaluate the functions for executing the cohesion tree, similarity tree, agnes and hclus algorithms are highly significant (p-value < 2.2e-16) and 2 to 2 comparisons is needed at de future.

We suggest that further research be conducted, where other operating systems are used, that more than 100000 data be used, the different methods, metrics and options (classic implication, classic implication+confidence, implifiance, euclidean, manhattan, ward.D, ward.D2, single, complete, average, mcquitty, median, centroid, aka, single, complete, ward, weighted, flexible, gaverage) are considered as factors, measure the occupied memory.

ACKNOWLEDGMENTS

We would like to thank the University of Salamanca Ph.D. Programme on Education in the Knowledge Society scope (http://knowledgesociety.usal.es) [29-31]. Similarly, we want to thank Escuela Superior Politécnica de Chimborazo for funding to perform this research.

REFERENCES

[1] Larry Johnson, Adams Becker, M. Cummins, Victoria Estrada, Alex Freeman, and C. Hall. 2016. *Horizon Report: 2016 Higher Education*.

[2] G. Siemens. 2013. Learning Analytics. The Emergence of a Discipline. *American Behavioral Scientist* 57, 10, 1380-1400. DOI:10.1177/0002764213498851.

[3] D. A. Gómez-Aguilar, F. J. García-Peñalvo, and R. Therón. 2014. Analítica Visual en eLearning. *El Profesional de la Información* 23, 3, 236-245. DOI:10.3145/epi.2014.may.03.

[4] D. A. Gómez-Aguilar, Á. Hernández-García, F. J. García-Peñalvo, and R. Therón. 2015. Tap into visual analysis of customization of grouping of activities in eLearning. *Computers in Human Behavior* 47, 60-67. DOI:http://dx.doi.org/10.1016/j.chb.2014.11.001.

[5] A. J. Berlanga and F. J. García-Peñalvo. 2005. IMS LD reusable elements for adaptive learning designs. *Journal of Interactive Media in Education* 11.

[6] A. J. Berlanga and F. J. García-Peñalvo. 2008. Learning Design in Adaptive Educational Hypermedia Systems. *Journal of Universal Computer Science* 14, 22, 3627-3647. DOI:10.3217/jucs-014-22-3627.

[7] D. Lerís and M. L. Sein-Echaluce. 2011. La personalización del aprendizaje: Un objetivo del paradigma educativo centrado en el aprendizaje. *Arbor* 187, Extra_3, 123-134. DOI:doi:10.3989/arbor.2011.Extra-3n3135.

[8] Tata. 2012. Learning Analytics in Enterprise Performance Management.

[9] Jacqueline Bichsel. 2012. *Analytics in higher education: Benefits, barriers, progress, and recommendations*. EDUCAUSE Center for Applied Research.

[10] Zacharoula K Papamitsiou and Anastasios A Economides. 2014. Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence. *Educational Technology & Society* 17, 4, 49-64.

[11] Ryan Shaun Baker and Paul Salvador Inventado. 2014. Educational Data Mining and Learning Analytics. In *Learning Analytics: From Research to Practice*, A.J. Larusson and B. White Eds. Springer New York, New York, NY, 61-75. DOI:10.1007/978-1-4614-3305-7_4.

[12] Rubén A Pazmiño-Maji, Francisco J García-Peñalvo, and Miguel A Conde-González. 2016. Approximation of statistical implicative analysis to learning analytics: a systematic review. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* ACM, 355-376.

[13] P. Michael, I. Elia, A. Gagatsis, and P. Kalogirou. 2010. Examining primary school students' operative apprehension of geometrical figures through a comparison between the hierarchical clustering of variables, implicative statistical analysis and confirmatory factor analysis *ASI5*, 19.

[14] Rubén Pazmiño, María Gabriela Pérez, and Victor Andaluz. 2014. Cuasi-implicación estadística y determinación automática de clases de equivalencia en imágenes de resonancia magnética de cerebro. *Revista Politécnica* 34, 1.

[15] Rubén A Pazmiño-Maji, Francisco J García-Peñalvo, and Miguel A Conde-González. 2017. Is it possible to apply Statistical Implicative Analysis in hierarchical cluster Analysis? First issues and answers. In *Congreso Internacional de Ciencia y Tecnología.*, P. Giade Ed., ESPOCH, Riobamba, Ecuador, 63-66.

[16] Cm Cuadras. 2012. Nuevos métodos de análisis multivariado. *Barcelona-España. CMC Editorial Monacor*.

[17] F. J. García, S. Bravo, M. Á. Conde, and H. Barbosa. 2011. SET, A CASE Tool to Guide the Creation of Domain and Use Case Models in an Introductory Software Engineering Course. *International Journal of Engineering Education (IJEE)* 27, 1, 31-40.

[18] Alboukadel Kassambara. 2017. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA.

[19] Ic Lerman, R Gras, and H Rostam. 1981. Élaboration et évaluation d'un indice d'implication pour des données binaires. 2. *Mathématiques et sciences humaines* 75, 5-47.

[20] Jean-Claude Regnier and Nadja Acioly-Regnier. 2007. Analyse cohésitive et interprétations des données dans le champ de l'éducation. In *Nouveaux apports à l'Analyse Statistique Implicative et Applications dans des Disciplines variées* Université JAUME 1 (Castellon) Espagne, 329-344.

[21] Raphaël Couturier. 2000. TRAITEMENT DE L'ANALYSE STATISTIQUE DANS CHIC. *ASI1*, 9.

[22] Nishikant Sonwalkar, J. Wilson, A. Ng, and P. B. Sloep. 2013. State-of-the-Field Discussion. *MOOCs Forum* 1, P, 6-9. DOI:10.1089/mooc.2013.0006.

[23] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[24] Donald T Campbell and Julian C Stanley. 2015. *Experimental and quasi-experimental designs for research*. Ravenio Books.

[25] Julian J Faraway. 2002. Practical regression and ANOVA using R University of Bath.

[26] Antonio Morillas. 2014. Muestreo en poblaciones finitas. *Apuntes: Muestreo*.

[27] Bas Giesbers, Bart Rienties, Dirk Tempelaar, and Wim Gijselaers. 2013. Investigating the relations between motivation, tool use, participation, and performance in an e-learning course using web-videoconferencing. *Computers in Human Behavior* 29, 1, 285-292.

[28] Frâncila Weidt Neiva, José Maria N David, Regina Braga, and Fernanda Campos. 2016. Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature. *Information and Software Technology* 72, 137-150.

[29] F. J. García-Peñalvo. 2014. Formación en la sociedad del conocimiento, un programa de doctorado con una perspectiva interdisciplinar. *Education in the Knowledge Society* 15, 1, 4-9.

[30] F. J. García-Peñalvo. 2015. Engineering contributions to a Knowledge Society multicultural perspective. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE RITA)* 10, 1, 17-18. DOI:10.1109/RITA.2015.2391371.

[31] F. J. García-Peñalvo. 2013. Education in knowledge society: A new PhD programme approach. In *Proceedings of the First International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013)*, F.J. García-Peñalvo Ed. ACM, New York, NY, USA, 575-577. DOI:http://dx.doi.org/10.1145/2536536.2536624.